

User Activity Classification for Adaptive Augmented Reality Task Guidance Systems

Samuel Lonneman

University of Florida

Undergraduate Honors Thesis

Bachelor of Science in Computer Science

May 2025, Summa Cum Laude

Abstract—This research explores the classification of user activity in the context of adaptive augmented reality (AR) task guidance systems. Building on a prior user modeling study, we use motion capture data collected from the head, hands, and eyes to predict high-level user activities including reading, searching, working, and idling. Activity labels were applied manually to video footage, then synchronized and mapped onto the motion data in order to train a classifier. We implemented a random forest model due to its simplicity and resistance to overfitting, before evaluating it using 5-fold cross-validation. The model achieved a high weighted average F1-score across all folds, with feature importance analysis highlighting the relevance of collected and derived attributes. Such a classifier is intended to be utilized to enable AR systems to adapt instructions in real time based on a user's current activity, potentially reducing cognitive load and improving task performance. While promising, the research is limited by a small set of labeled data and lacks sophisticated temporal modeling. Future work will focus on expanding the dataset, improving generalization, and incorporating more advanced time series models. These findings represent a significant step towards more personalized and responsive AR systems for education, manufacturing, and beyond.

1. Introduction

The primary goal of task guidance systems is to assist users in completing complex tasks by providing step-by-step instructions. These systems seek to reduce errors, improve user efficiency, and improve learning by offering real-time support tailored to the task at hand [1]–[3]. Traditional forms of task guidance include printed manuals, static digital instructions, and prerecorded videos, all of which have limitations in terms of interactivity and adaptability. As tasks become more complex and intricate, whether in industrial assembly, medical procedures, or educational training,

there is an increased need for more responsive and dynamic guidance systems that can adapt to user behavior and variations in task execution.

A modern method for the delivery of task guidance is Augmented Reality (AR) technology, where digital information is presented as an overlay onto the physical environment of the user [4]–[6]. This is often accomplished using a handheld device like a smartphone with a camera or via AR headsets like Microsoft's HoloLens or Apple's Vision Pro. Using AR technology allows for real-time visual and spatial cues, which provides a contextualized and seamless

user experience. Furthermore, by integrating the instructions directly into the user's view, it can help a user to avoid external distractions and remain focused on the task at hand. AR headsets in particular allow for hands-free operation, enabling the user to continue working while viewing instructions. Eliminating this context switch can help reduce the cognitive load required by the user when compared with traditional task guidance methods [4], [7].

There are a number of applications in which AR task guidance systems are useful. In manufacturing contexts, such systems have been adopted for industrial and mechanical assembly tasks as a key part of "Industry 4.0," aligned with the concept of smart manufacturing and interconnected systems [7]–[9]. In training and education, AR task guidance systems allow for hands-on learning experience for students and professionals alike [4], [6], [10]. Finally, in maintenance and repair tasks, such systems can assist in providing real-time diagnostics and repair instructions in situ [8], [10].

Several challenges exist with static guidance systems. First, tasks can be complex with multiple potential pathways to completion. Furthermore, users may deviate from predefined steps or alter their working environment by moving objects. The cognitive state of the user can also be an important factor. Users can get stuck, become overwhelmed, exhausted, or distracted due to elements related to or external to the task. For these reasons, a system is required which can not only adjust to what the user is doing, but also to their cognitive load [7].

In order to develop such a system, real-time data is needed for determining how the user is feeling and what the user is currently doing, so that it may be compared to what the user needs to do to complete their task. A prior user modeling study [11] collected data from various aspects of the user's experience primarily for the purpose of understanding a user's cognitive load. Specifically, the study collected pupilometry data and physiological data including body temperature, the galvanic skin response, blood volume pulse, heart rate, and inter-beat interval. In addition to this data relevant to cognitive state, spatial tracking data was also collected. Using motion capture technology, the physical position and rotation of a user's head and hands were recorded throughout the study. This spatial data is of particular interest to this study, which will focus on classifying what type of activity the user is engaged in at a given moment.

Thus, this research aims to develop a method for

classifying a user's activity at a given moment based on physiological sensor data. This could prove useful in future work on adaptive task guidance systems since the current activity can be compared to the recommended activity to allow the system to adapt accordingly. Such a system may adjust instruction complexity or pacing, provide additional prompts or demonstrations when needed, and offer assistance when a user is stuck. When combined with appropriate detection of a user's cognitive state, a system would be even more apt at meeting the needs of the user.

In all, the work of this thesis consists of the following:

- Assist in the collection of physiological and behavioral data from a user study in the context of an AR assembly procedure.
- Manually review video recordings and label user activity synchronized with the collected sensor data.
- Analyze labeled data and create a classifier to predict a user's activity at a given moment based on physiological sensor data.
- Evaluate the performance of the classifier and provide recommendations for future improvements and applications.

2. Methods

Toward the goal of creating a user activity classifier, this section will discuss how individual user activity types are identified, labeled, analyzed, and classified based on the available data, as well as why exploring why this is valuable. All code used for pre-processing, model training, and evaluation is available in a public GitHub repository at <https://github.com/SamLonneman/User-Activity-Classification-for-AR>.

2.1. Data Collection

As described in the introduction, the classifier was designed based on data collected in a prior study which produced the Multi-Modal User Modeling for Task Guidance (MUMTG) Dataset [11]. This dataset includes log data from 20 undergraduate student participants as they are guided through search and assembly tasks in a game-like scenario, as depicted in Figure 1. In particular, it includes three main types of data: details related to the state of the game, motion tracking data from hand and head-worn sensors, physiological data, and self-reported ratings related to the workload of tasks and emotional response. Almost every datatype is captured several times per second and

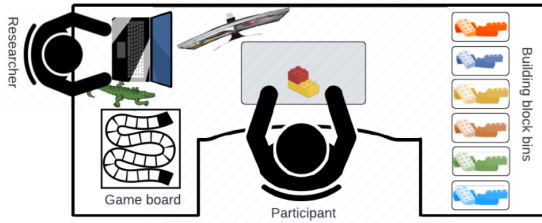


FIGURE 1: The MUMTG game-like environment seats a participant at a table where they follow instructions to complete assembly tasks using building blocks. The story unfolds on the game board, narrated by the researcher. Image from Hashky et. al. [12] used with author permission.

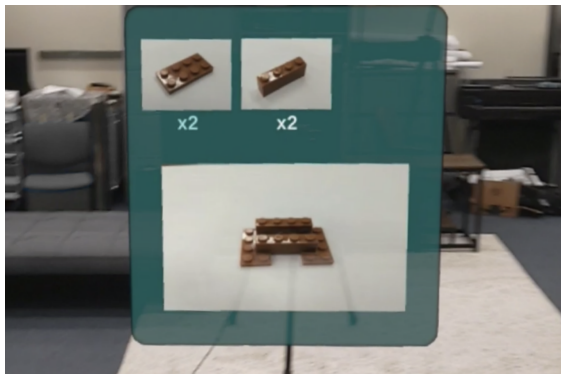


FIGURE 2: Augmented Reality delivery of instructions. This step tells the user to collect 2 of each pictured block, to be attached in a specific arrangement. Image from Hashky et. al. [12] used with author permission.

recorded throughout the duration of the study, except for the survey data, which is recorded periodically between major segments of the study and retroactively applied to each sample. For the purpose of determining the type of activity being performed at any moment, we have chosen to focus primarily on the head and hand tracking data. Data such as heart rate, temperature, electrodermal activity, emotional response, and state of the game are less meaningful on short timescales and will not be considered.

2.2. Justification for Activity Labeling

In order to produce a highly adaptive and effective task guidance system, it is valuable to understand precisely what a user is attempting to accomplish at a given moment. When combined with knowledge of what a user needs to be doing in order to progress, a

task guidance system can provide relevant feedback or otherwise adjust guidance to best suit the user. There are a variety of challenges a user can face during a task, leading to variability in execution. For example, a user may pause, get distracted, switch tasks, deviate from expected actions, or get stuck. In order to capture these nuances in user behavior, activities from this data must be labeled to provide a ground truth for training and evaluating an activity classification model.

2.3. Activity Labeling Process

Fortunately, the MUMTG study included video recordings of participants during the completion of tasks. By watching these videos of participant behavior, the following high level activity types were identified, and they are summarized below in Table 1. These activities collectively capture every moment of a user’s behavior during the MUMTG study. To give context about how each of these activity types fit together, we will walk through an example scenario. To start, a user is presented with instructions in their AR headset or via the monitor in their workspace. The user gazes at the instructions with hands and head stationary, so we label the segment “Reading.” The user understands they need to find two red building blocks to begin their task, so they begin to turn their head around, visually scanning their workplace for the container with red blocks. We would label this “Searching.” The user then reaches out with their hand(s), grabbing the container and bringing it closer to their body, so we label this segment “Retrieving.” The user uses their hands to shuffle blocks around in the container, trying to find the particular red blocks they need. We label this segment “Rummaging.” The user uses their hands to manipulate the blocks, connecting them together. We label this segment “Working.” The user has now completed the task, and sits idly, listening to the facilitator explain how to fill out the survey. We label this segment “Idling.” The user begins actively filling out the survey on laptop in their workspace, so finally we label this segment “Survey.” These events may occur in any order and for varying lengths of time, so it is crucial that we developed a streamlined method for performing this labeling.

In order to label the individual activities that are being completed, we utilized the qualitative data analysis software MAXQDA. With this software, we are able to place labels on segments of a video as shown in Figure 4, before exporting a detailed list of labels and their corresponding intervals in time.

Activity Type	Description
Reading	User is looking at the provided task instructions.
Searching	User is visually scanning the workspace for an object of interest.
Retrieving	User is physically reaching into their environment for an object or container.
Rummaging	User is moving items within a container to locate and obtain an object of interest.
Working	User is using fine motor skills to manipulate objects in their hands.
Idling	User appears to be doing nothing.
Survey	User is completing a survey on a laptop.

TABLE 1: Activity types and their descriptions.

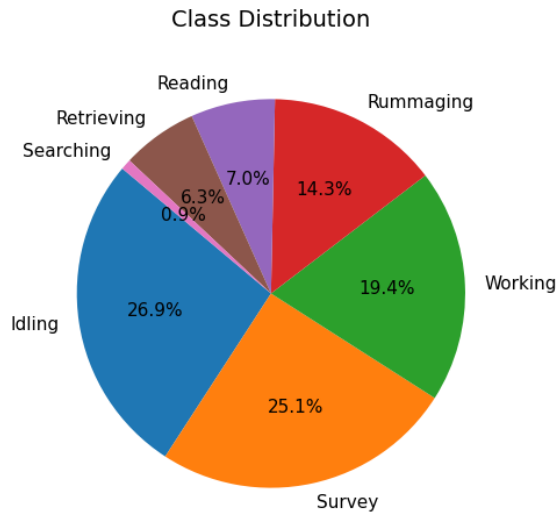


FIGURE 3: Class Distribution. This chart shows the proportion of all timestamps labeled with each activity type. Notice that the class distribution is imbalanced, with “Idling” and “Searching” dominating, and “Searching” and “Retrieving” in the minority. Special care must be taken to properly train a model on this imbalanced dataset.

After collecting all relevant activity labels for each participant, we mapped them onto the existing motion tracking data logged during the session, creating a new attribute “Activity” for each timestep. Each labeled timestep will serve as an example (an instance of user behavior) with which a model can be trained. Figure 3 displays the resulting distribution of classes (activity types) for the labeled timestamps.

2.4. Motion Data Processing

There are multiple sources of motion tracking data that we are interested in. First, the head-worn Mi-

crosoft HoloLens 2 augmented reality headset reports the current position and rotation of the participant’s head in space. It also uses onboard cameras and computer vision to track the position and rotation of the user’s hands in the same coordinate space. For redundancy, the participant also wore gloves with attached reflective markers, observed by an OptiTrack 13W infrared camera system to determine positions and rotations of the participant’s hands. All of these positions and rotations are collected multiple times per second and recorded as their X, Y, and Z components respectively. At any moment during the study, it is possible for the HoloLens or OptiTrack system to lose sight of the participant’s hands. In these moments, they simply report the last position and rotation that they had sight of the hands. While this is convenient and avoids gaps in the logs, it can introduce inaccuracies when the hands are lost for an extended period. For this reason, we filter out any entries where this phenomenon occurs.

2.5. Feature Engineering

For the purposes of exploratory data analysis and development of a classifier, it is helpful to engineer additional features from the motion tracking data provided. To create a sense of temporal context beyond a single frame, we calculated new “speed” attributes for the head and hands, based on their displacement from the prior frame. We also calculated and created new attributes representing the distance between the two hands, as well as the distance of each head/hand to its average position. This gives insight into when the user is exhibiting extreme behavior, such as reaching far away from their body or workspace.

2.6. Activity Classification Model

After labeling, preprocessing, and engineering new features for the data, we were able to begin creating a model for activity classification. The remaining sections will describe the exploratory data analysis, selection, development, and evaluation for the model.

2.6.1. Model Selection Initial plans for the classifier involved performing exploratory data analysis to uncover underlying relationships between the motion tracked features and the corresponding activity labels, before manually drawing thresholds and developing a rule-based heuristic algorithm. While early analysis showed some clear and promising distinctions between certain activity types, others were largely



FIGURE 4: Labeling activity types in video recordings using MAXQDA. This tedious process is necessary to develop a large set of known examples in order to train and validate the activity classifier.

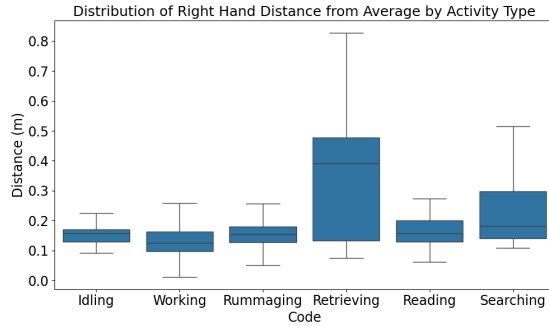


FIGURE 5: Distribution of right hand distance from average by activity type. Notice that this metric is clearly associated with the retrieval activity, but is inconclusive regarding the difference between reading and working.

indistinguishable. For example, Figure 5 demonstrates that a high right hand distance from average can be clearly associated with the retrieval activity. However, this attribute indicates little if anything regarding the difference between reading and working. Due to the difficulty imposed by manually developing a rule-based heuristic, we decided to proceed using a simple machine learning model instead.

In an effort to create a highly interpretable model, we initially chose to train a decision tree classifier model due to their high interpretability. Once trained, a decision tree provides a precise flowchart-like structure where each node corresponds to a feature, and each branch represents a decision based on the value of that feature [13]. For example, this allows us to trace how specific feature thresholds (e.g., right hand displacement greater than 2 feet) is associated with an activity such as retrieval. Finally, properly tuned decision trees are well equipped to mitigate overfitting with established techniques like pruning, setting maximum depth or number of leaf nodes, and extending to ensemble methods like random forests. To begin, we split the data into two subsets for training (80%) and testing (20%). Naive implementations for training a decision tree initially appeared highly successful, with

extremely high accuracy and F1-score on the held-out test set. However, further investigation revealed that this is likely due to overfitting. In this case, a very deep decision tree has the capability to memorize each occurrence of an activity, rather than generalizing to activities as a class. In order to combat this, we ultimately decided to train a random forest model, intended to be more resistant to overfitting [13]. A random forest builds on the capabilities of a decision tree, essentially fitting many trees to randomly selected subsets of the training data, combining them into a committee which votes on the classification of a given sample [14]. This inherently reduces the potential for overfitting, making it a more robust method for this use case.

2.6.2. Model Training and Evaluation In order to train the random forest classifier, we relied on Scikit-learn, a user friendly machine learning library for Python [15]. Towards the effort to create a model resistant to overfitting, we implemented 5-fold stratified cross validation. To do so, we divide the entire dataset into 5 equal (or near equal) subsets called folds. On each iteration, we train the model on 4 of the folds, leaving one out as a test set. The process repeats 5 times, with each fold used exactly once as the test set. Stratification ensures that the distribution of class labels (e.g. idling, Retrieval, Working) is roughly equal in each fold as it is in the full dataset. This is especially important since we have an imbalanced dataset as shown in Figure 3.

The final evaluation metric is F1-score, and it is averaged across all 5 folds. This helps provide a more reliable performance estimate and gives a better sense of how the model is expected to perform on unseen data than a single train/test split. Another way we seek to minimize the effects of class imbalance in the training data is to employ the `class_weight='balanced'` parameter in the decision tree classifier. This approach adjusts the weights of each class inversely proportional to their frequency in the training set, making sure that underrepresented

classes receive more attention during training. Finally, in order to reduce overfitting, the maximum number of trees in the voting committee was set to 100, and the maximum number of leaf nodes on each tree was set to 100. This helps to ensure that the random forest is able to generalize to unseen data, rather than simply memorizing the training data.

3. Results

To evaluate the performance of the activity classification model, we conducted 5-fold stratified cross validation on the dataset of labeled participant logs. The section below presents the results of this cross validation, providing a robust estimate of generalization performance.

3.1. Quantitative Evaluation

For each fold of the 5-fold stratified cross validation, a classification report was generated, including the precision, recall, F1-score, and support for each of the 7 labeled activity types. The classification accuracy was also calculated, as well as the macro and weighted averages of each metric. An example of the classification report for a single fold is provided in Table 2.

Activity	Precision	Recall	F1-Score	Support
Reading	0.55	0.82	0.66	1822
Searching	0.71	0.99	0.83	230
Retrieving	0.73	0.86	0.79	1637
Rummaging	0.79	0.78	0.78	3711
Working	0.90	0.72	0.80	5043
Idling	0.99	0.96	0.98	6994
Survey	0.99	0.99	0.99	6524
Accuracy			0.88	25961
Macro Avg	0.81	0.88	0.83	25961
Wtd Avg	0.90	0.88	0.88	25961
Std Dev	0.16	0.11	0.12	2597

TABLE 2: Classification Report for one cross validation fold. Observe the F1-scores for each activity to notice that the model underperforms on minority classes while performing very well for dominant classes. This indicates future work is required to better manage the inherent class imbalance.

Likewise, a confusion matrix heat map was calculated for each fold, summarizing the model's predictions and highlighting how well the model performs for each class and where it makes mistakes. The confusion matrix for the same fold is given in Figure 6.

Once all folds are complete, the average of the

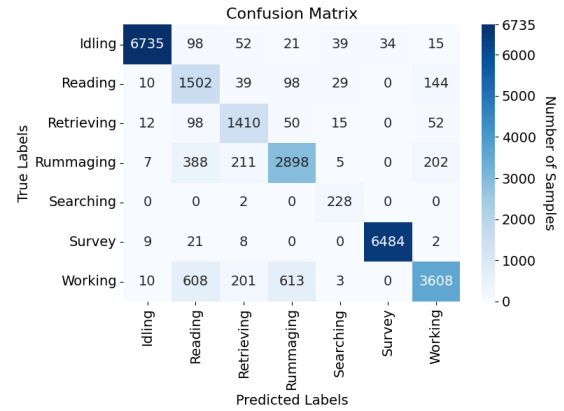


FIGURE 6: This confusion matrix represents the true labels vs those predicted by the model to demonstrate precisely where the model is correctly and incorrectly classifying examples. The visible diagonal line represents correct classifications, with color off the diagonal representing misclassifications. Notice that the working activity was sometimes confused with the reading, retrieving, and rummaging activities.

weighted average F1-score for each fold was calculated, providing an estimate of generalization performance. These results are provided in Table 3.

Fold	Weighted Average F1-Score
1	0.88
2	0.88
3	0.88
4	0.89
5	0.88
Average	0.882
Standard Deviation	0.004

TABLE 3: Results of 5-Fold Stratified Cross Validation. Notice that each fold has nearly identical performance as measured by the weighted average F1-score.

3.2. Feature Importance

As part of the model evaluation, we investigated the relative importance of the many different features in predicting user activity. Feature importance indicates how directly each attribute contributes to the decision making process of the random forest model, with higher values reflecting stronger contributions. To derive these scores, we used the built in feature importance attribute of Scikit-learn's random forest classifier. A bar chart showing the importance of each

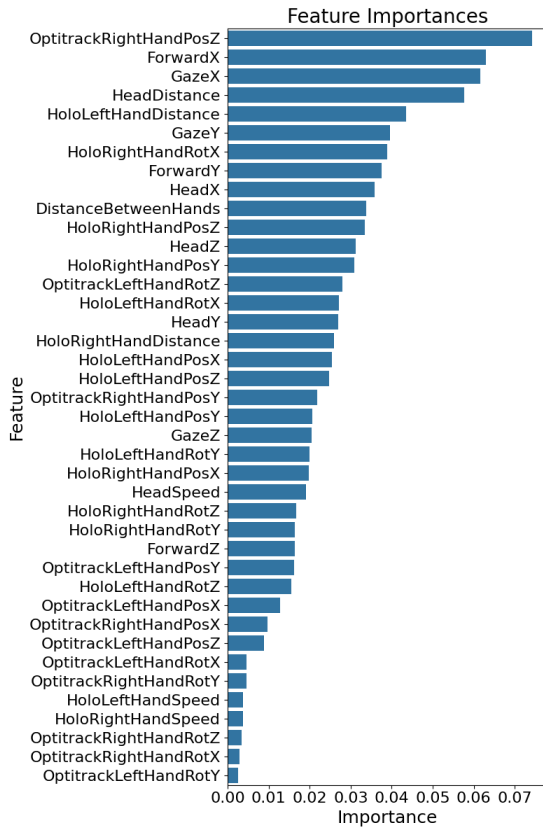


FIGURE 7: Random Forest Feature Importances. This chart gives the relative importance of each feature. Notice that the top three features are from the hands, head, and eyes, indicating that all of these sensors are valuable to the classifier.

feature is presented in Figure 7.

4. Discussion

We begin with a summary of major findings. First, Table 3 demonstrates a high weighted average F1-score on all five folds of the stratified cross validation. The average across folds of 0.882 indicates a strong balance between precision (how many of the predicted positives are actually positive) and recall (how many of the actual positives are correctly identified), suggesting that the model performs well in classifying user activity according to the provided attributes. As described in the methods section, the random forest model was trained with balanced class weighting to address the inherent imbalance in the training data. However, as reflected in the per-class F1-scores in Table 2, performance on minority classes remained lower than for dominant classes. This indicates that

while class weighting may have mitigated some bias, further efforts will be required to achieve fully balanced performance across the different activity types, such as oversampling or undersampling.

4.1. Interpretation of Feature Importance

As previously described, a key benefit of random forest models is their ability to estimate feature importance, giving insight into which attributes the model relies on most when making predictions. As shown in Figure 7, the top contributing feature was `OptitrackRightHandPosZ`, followed closely by `ForwardX`, `GazeX`, and `HeadDistance`. This indicates that positional data from the hands, head, and eyes play a central role in distinguishing between user activities. For example, the vertical position of the right hand `OptitrackRightHandPosZ` can be used to determine whether a user is working (usually with hands at the level of the table), reaching (hands reach up and out into environment), or idling (hands down in lap), depending on the task context. Also notice that the composite features `HeadDistance`, `HoloLeftHandDistance`, and `DistanceBetweenHands` were also highly ranked, indicating that relative motion and spatial relationships between body parts are useful in characterizing behavior. In contrast, many of the rotational features (especially those collected by the OptiTrack system) appeared much lower in the list, suggesting that angular orientation of hands may be less informative in this task setting. Finally, the speed-related derived features `HoloLeftHandSpeed`, `HoloRightHandSpeed` and `HeadSpeed` received low importance scores, indicating that task classification relies more on posture than movement. Overall, this analysis helps to interpret the internal logic of the model. Perhaps even more importantly, it supports the broader claim that physical activity types can be meaningfully characterized by motion capture data, especially when considering spatial relationships and posture.

4.2. Error Analysis

The confusion matrix given in Figure 6 provides valuable insight into the specific correct and incorrect predictions made by the model, highlighting both its strengths and its weaknesses in the form of its classification patterns. Notice that the model achieves high accuracy across most activity types, as indicated by the highlighting along the main diagonal. The activities most often classified correctly are Idling,

Survey, and Working, with a substantial number of true positives. However, some activities exhibit notable confusion. Take for example the Working activity, which is frequently confused with Reading (608 instances), Retrieving (201), and Rummaging (613). This suggests that the model struggles to differentiate subtle differences in motion patterns, possibly due to similarities in the gestures made during these activities.

4.3. Limitations

While this research has been fruitful, the progress thus far has largely been a work-in-progress exploration of methods and practical challenges. We therefore admit many limitations. Most notably, we used a very limited set of labeled data, only considering data from two participants. As a result, the model may not generalize well to broader populations, especially those with different physical characteristics and movement styles. Furthermore, the participants were observed and measured in a highly controlled environment. While this well-structured setup includes elements of disruption and surprise [12], it may not fully reflect the unpredictability of real-world settings. Activities were labeled manually by a single researcher and may contain subjective biases and inaccuracies. Another important limitation is that the current model treats each data point as an independent example without explicitly modeling time-series dependencies (besides a crude calculation of instantaneous object speed). Thus, the model does not consider the more complex and potentially valuable temporal patterns like continuous motion paths. Finally, it should be noted that there may exist some dependence between the train and test sets since the current method of record-wise random selection can cause nearly identical records from a single labeled segment to appear in both sets, inflating the model's performance on the test set.

4.4. Future Research

To address the limitation of the labeled dataset being very small and only considering data from two participants, future research could include continuing data labeling for many more participants from the original dataset [11]. This could help to improve classification performance in general by providing more labeled examples. The current method for labeling data is relatively tedious and slow, making it difficult to label vast amounts of data at scale. Further methods could be designed to speed up this process, and it could be distributed among multiple people to

assist in scaling. Not only would more labeled data help to improve classification performance, it could help improve the generalization across participants, ensuring that the model isn't overfitted to a particular participant's behavior patterns. Beyond ordinary stratified cross validation, future research could implement Leave-One-Subject-Out Cross-Validation (LOSOCV). Instead of combining and shuffling the data from all participants and dividing it into folds for training and evaluation, each LOSOCV fold holds out all data from a single participant for evaluation. This method gives a subject-independent estimation of performance for unseen participants [16], helping to better judge how the model will perform in practicality. Other avenues for future research include moving beyond random forests to more advanced models, particularly those well suited to learning patterns from time-series data like hidden Markov models (HMMs) [17] or recurrent neural networks (RNNs) [18]. Finally, to address the potential dependence between the train and test sets, future work could include replacing the current record-wise random selection with segment-wise random selection. This would ensure that all records from a given segment appear in only the train set or only the test set, helping to guarantee independence and better estimate generalization performance.

5. Conclusion

This research sought to develop an automated method for the classification of user activities while engaging with an augmented reality task guidance system, using features derived from the motion capture data from multiple participants. By focusing on high-level abstract activities such as reading, searching, working, and idling, we aimed to contribute towards improving the adaptability and effectiveness of AR systems by predicting user actions in real time. The results demonstrated that the random forest classifier performed well in predicting these activity types, with a clear report of key features that influenced the classification accuracy. Notably, the normalized motion capture data from hand, head, and eye tracking devices all proved important in distinguishing between activity types. These findings contribute to the understanding of how sensor data from motion capture gloves and head worn devices can be used to enhance AR task support systems. This research has wide ranging implications, particularly for the development of adaptive and responsive AR systems. Through the integration

of effective real-time activity classification, an AR system will be able to adjust its support based on a user's current activity, improving task performance and lowering cognitive load. This work opens the door for further research, with potential applications in education, manufacturing, and even healthcare. While this study provides some valuable insights, it is not without its limitations, such as the small number of labeled participant data logs used to train the model, the lack of temporal pattern recognition, and the potential for subjective biases in the labeling process. Future work could explore broader datasets, use more advanced models and validation techniques, and integrate additional sensor modalities to improve the classifier's generalization abilities. In conclusion, the research presented here demonstrates the potential for machine learning-based activity classification to improve AR task guidance systems, paving the way for more adaptive and personalized user experiences in a variety of real-world applications.

■ REFERENCES

1. M. Ryoo, K. Grauman, and J. Aggarwal, "A task-driven intelligent workspace system to provide guidance feedback," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 520–534, 2010. Special issue on Intelligent Vision Systems.
2. N. L. Kirsch, S. P. Levine, R. Lajiness-O'Neill, and M. Schnyder, "Computer-assisted interactive task guidance: Facilitating the performance of a simulated vocational task," *The Journal of Head Trauma Rehabilitation*, vol. 7, no. 3, 1992.
3. G. Rudnick and A. Schulte, "Implementation of a responsive human automation interaction concept for task-based-guidance systems," in *Engineering Psychology and Cognitive Ergonomics: Performance, Emotion and Situation Awareness* (D. Harris, ed.), (Cham), pp. 394–405, Springer International Publishing, 2017.
4. M. Weerasinghe, A. Quigley, K. Pucihar, A. Toniolo, A. Miguel, and M. Kljun, "Arigatō: Effects of adaptive guidance on engagement and performance in augmented reality learning environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3737–3747, 2022.
5. A. Sajadi, M. Alaeifard, and M. Modaberi, "Argus: Visualization of ai and machine learning-assisted task guidance in augmented reality," Jan. 2025. Available at SSRN: <https://ssrn.com/abstract=5094311>.
6. G. Huang, X. Qian, T. Wang, F. Patel, M. Sreeram, Y. Cao, K. Ramani, and A. J. Quinn, "Adaptatur: An adaptive tutoring system for machine tasks in augmented reality," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.
7. X. Wang, S. Ong, and A. Nee, "Multi-modal augmented-reality assembly guidance based on bare-hand interface," *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 406–421, 2016.
8. M. L. Yuan, S. K. Ong, and A. Y. C. N. and, "Augmented reality for assembly guidance using a virtual interactive tool," *International Journal of Production Research*, vol. 46, no. 7, pp. 1745–1767, 2008.
9. G. Mompeu, F. Danglade, F. Mérienne, and C. Guillet, "Methodology for augmented reality-based adaptive assistance in industry," *Computers in Industry*, vol. 154, p. 104021, 2024.
10. C. Truong-Allié, A. Paljic, A. Roux, and M. Herbeth, "User behavior adaptive ar guidance for wayfinding and tasks completion," *Multimodal Technologies and Interaction*, vol. 5, no. 11, 2021.
11. A. Hashky, B. Rheault, A. R. Ahsan, B. Benda, T. Audino, S. Lonneman, and E. Ragan, "Multi-modal user modeling for task guidance (mutmg): A dataset for real-time assistance with stress and interruption dynamics." Mendeley Data, V1, 2024.
12. A. Hashky, B. Rheault, A. R. Ahsan, B. Benda, T. Audino, S. Lonneman, and E. D. Ragan, "Multi-modal user modeling for task guidance: A dataset for real-time assistance with stress and interruption dynamics," in *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 544–550, 2024.
13. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, Springer New York, NY, 2 ed., 2009.
14. L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.
15. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, and G. Louppe, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, 01 2012.
16. D. Gholamiangonabadi and K. Grolinger, "Personalized models for human activity recognition with wearable

sensors: deep neural networks and signal processing,” *Applied Intelligence*, vol. 53, p. 6041–6061, July 2022.

17. B. Mor, S. Garhwal, and A. Loura, “A systematic review of hidden markov models and their applications,” *Archives of Computational Methods in Engineering*, vol. 28, 05 2020.
18. H. Wang and L. Wang, “ Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks ,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 3633–3642, IEEE Computer Society, July 2017.

Samuel Lonneman is an undergraduate researcher in the Indie Lab (Interactive Data and Immersive Environments) at the University of Florida, advised by Eric D. Ragan, Ph.D. His current research interests include virtual and augmented reality and adaptive systems. Contact him at lonneman.sam@gmail.com.